

Those who are enamored of practice without theory are like a pilot who goes into a ship without rudder or compass and never has any certainty where he is going. Practice should always be based on a sound knowledge of theory.

Leonardo Da Vinci (1452-1519)



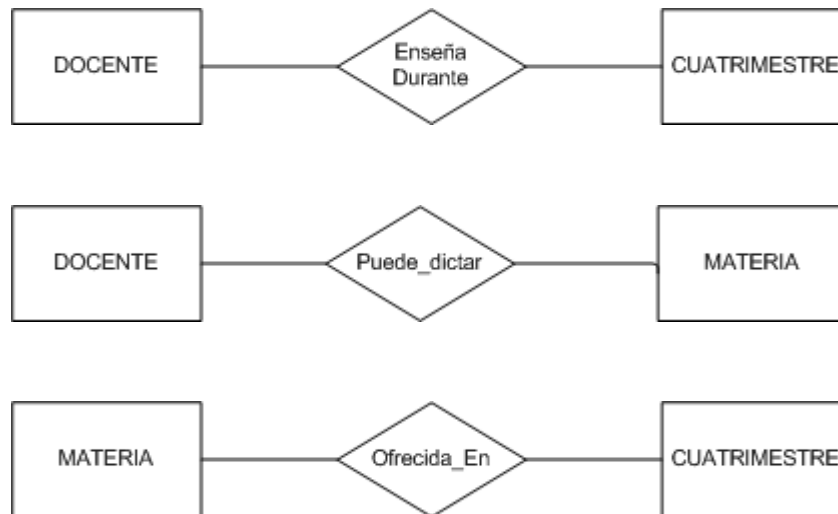
MER –MR Bases de Datos



Lic. Gerardo Rossel

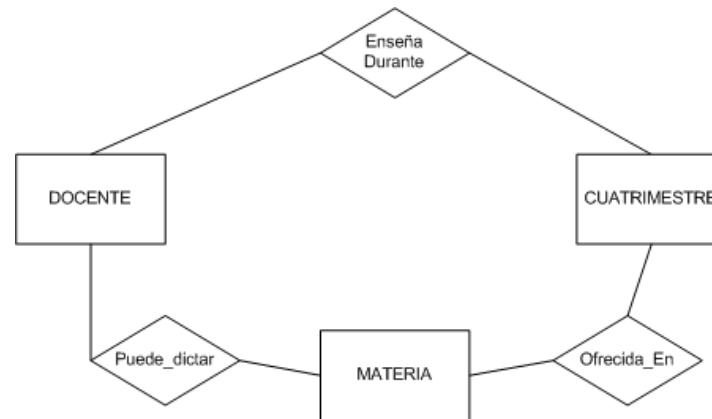
Interrelaciones Ternarias

- Los **docentes** pueden dictar potencialmente varias materias según su especialidad. Las **materias** son ofrecidas durante **cuatrimestres**, a su vez es posible que diferentes docentes dicten la misma materia en distintos cuatrimestres. Los docentes dictan sólo una materia por cuatrimestre

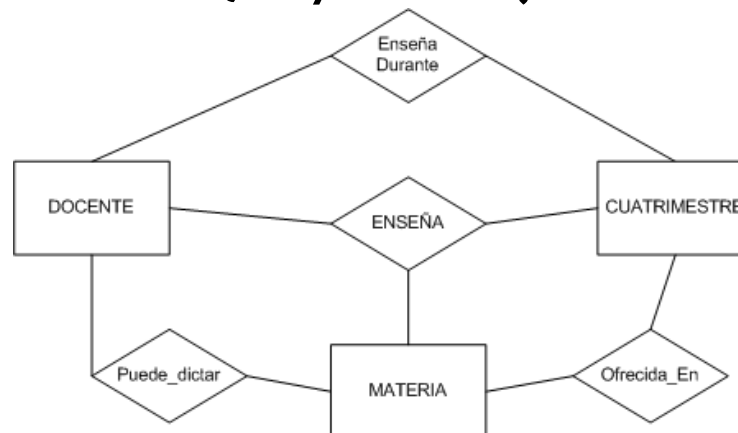


Interrelaciones Ternarias

► Quedaría:

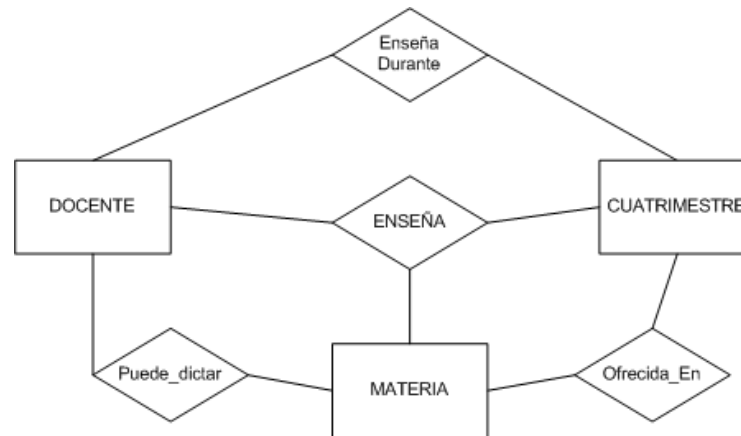


- ¿Como capturamos la información: *José enseñó Base de Datos en el 1er Cuat de 1967?* ¿Hay un mejor modelo?



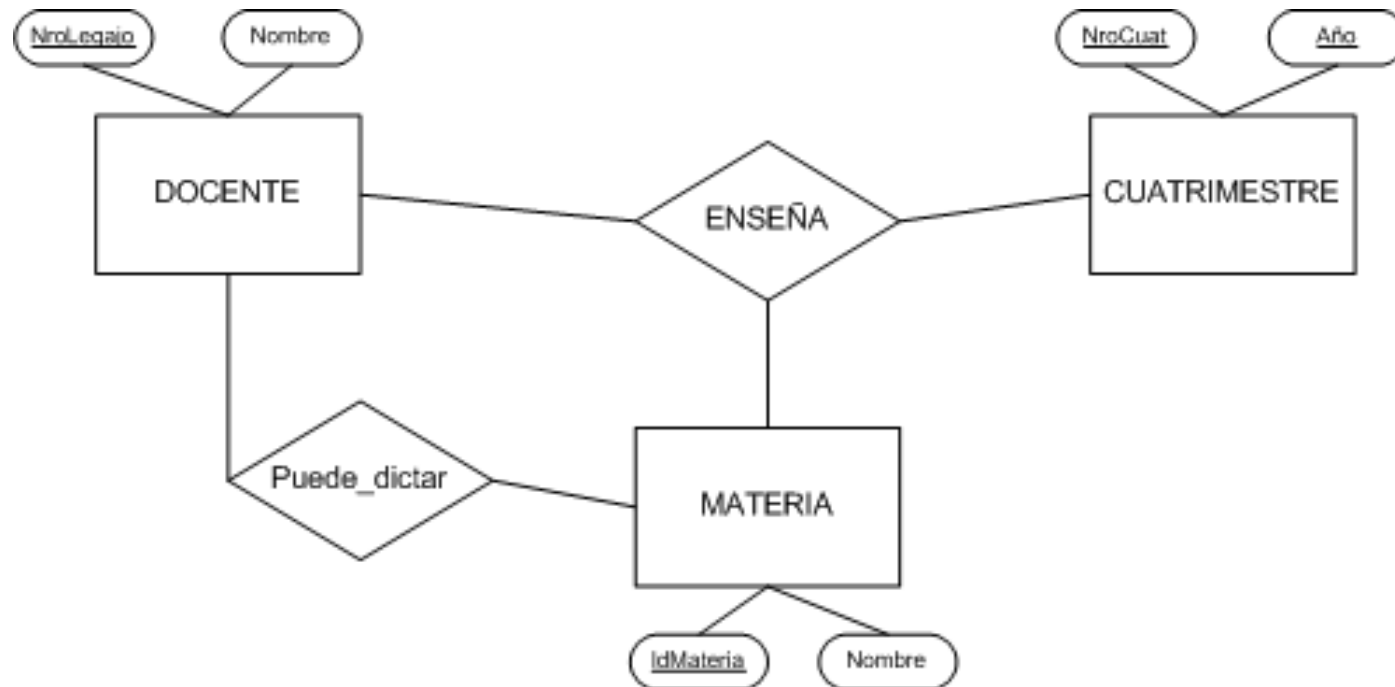
Interrelaciones Ternarias

► Interrelaciones Ternarias



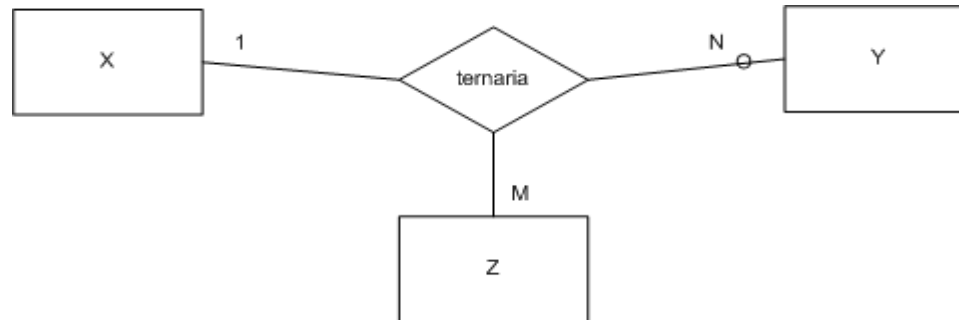
- Una instancia (d,c,m) en **Enseña** no existiría si no existieran instancias (d,c) en **Enseña_Durante**, (c,m) en **Ofrecida_En** y (d,c) en **Puede_Dictar**.
- La reversa no es válida.
- Se pueden deducir las instancias de **Enseña_Durante** y **Ofrecida_En** a partir de Enseña (REDUNDANTES).

Interrelaciones Ternarias



Interrelaciones Ternarias

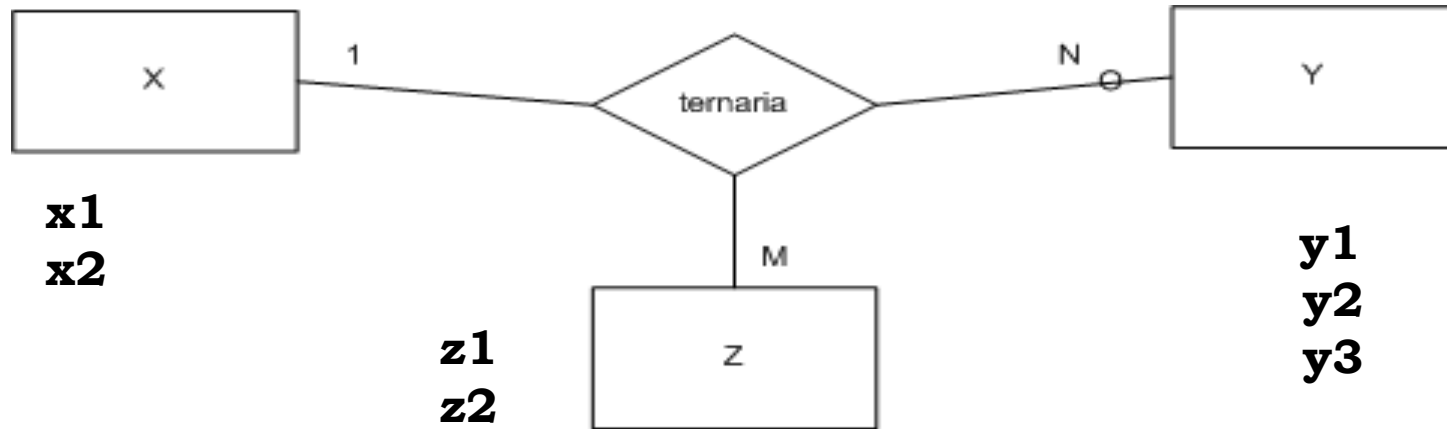
- ▶ La **cardinalidad** se define tomando de a dos entidades.
 - ▶ Un par (x,z) puede estar relacionado con muchos y's
 - ▶ Un par (x,y) puede estar relacionado con muchos z's
 - ▶ Un par (y,z) puede estar relacionado con un solo x



- ▶ La **participación** se define individualmente
 - ▶ X participa totalmente en la ternaria
 - ▶ Y participa parcialmente en la ternaria
 - ▶ Z participa totalmente en la ternaria

Interrelaciones Ternarias

► Restricciones, Cardinalidad y Participación



VÁLIDA

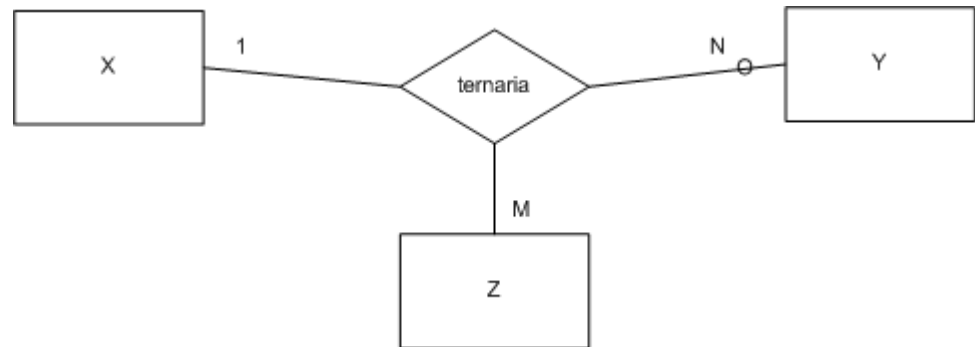
<x1,y1,z1>
<x1,y1,z2>
<x2,y2,z2>

INVÁLIDAS

<x1,y1,z1>
<x1,y1,z2>
<x1,y2,z2>
<x2,y2,z2>

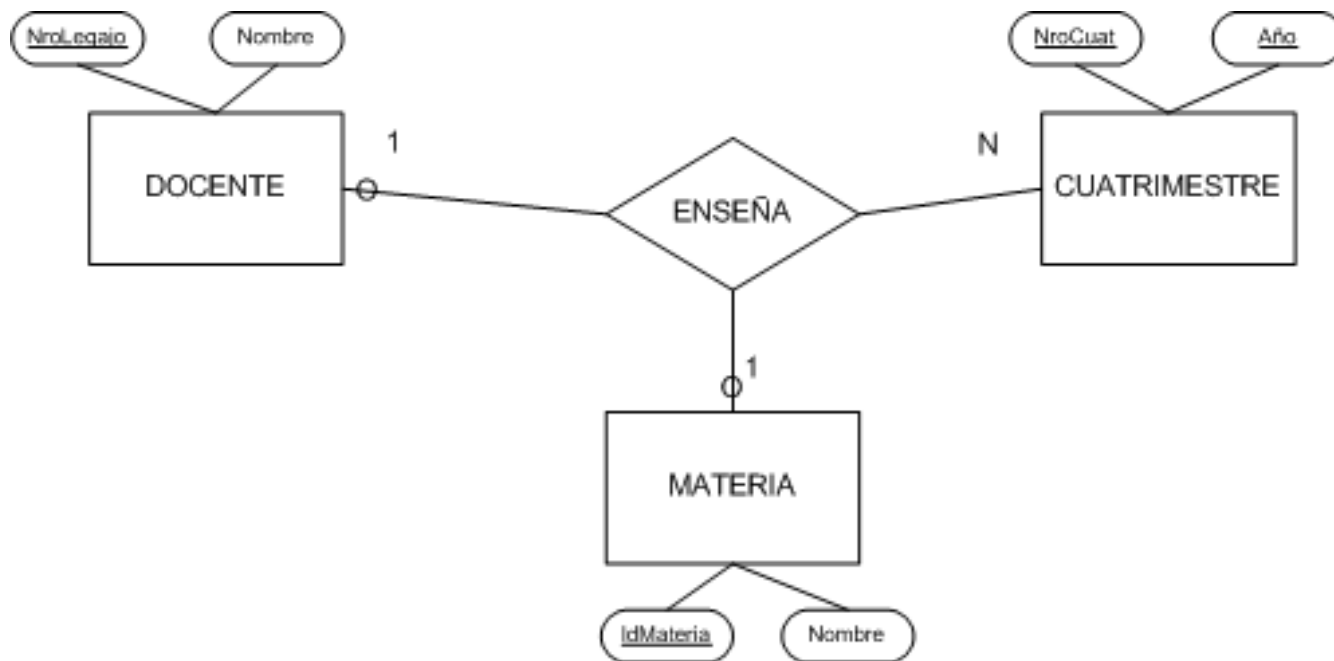
Interrelaciones Ternarias

- ▶ Un elemento de la ternaria **SIEMPRE** tendrá un **elemento de cada entidad participante**
- ▶ Es decir, en la relación ternaria podremos tener:
 - ▶ $\langle x1, y2, z3 \rangle$
 - ▶ $\langle x2, y2, z3 \rangle$
- ▶ Pero **NUNCA**
 - ▶ $\langle x1, \text{null}, z2 \rangle$



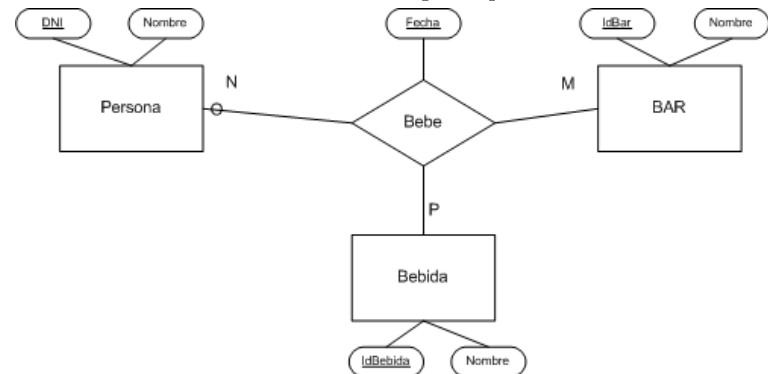
Interrelaciones Ternarias

- Agregamos entonces cardinalidad y participación
 - Estas restricciones dependen del espacio del problema en concreto.

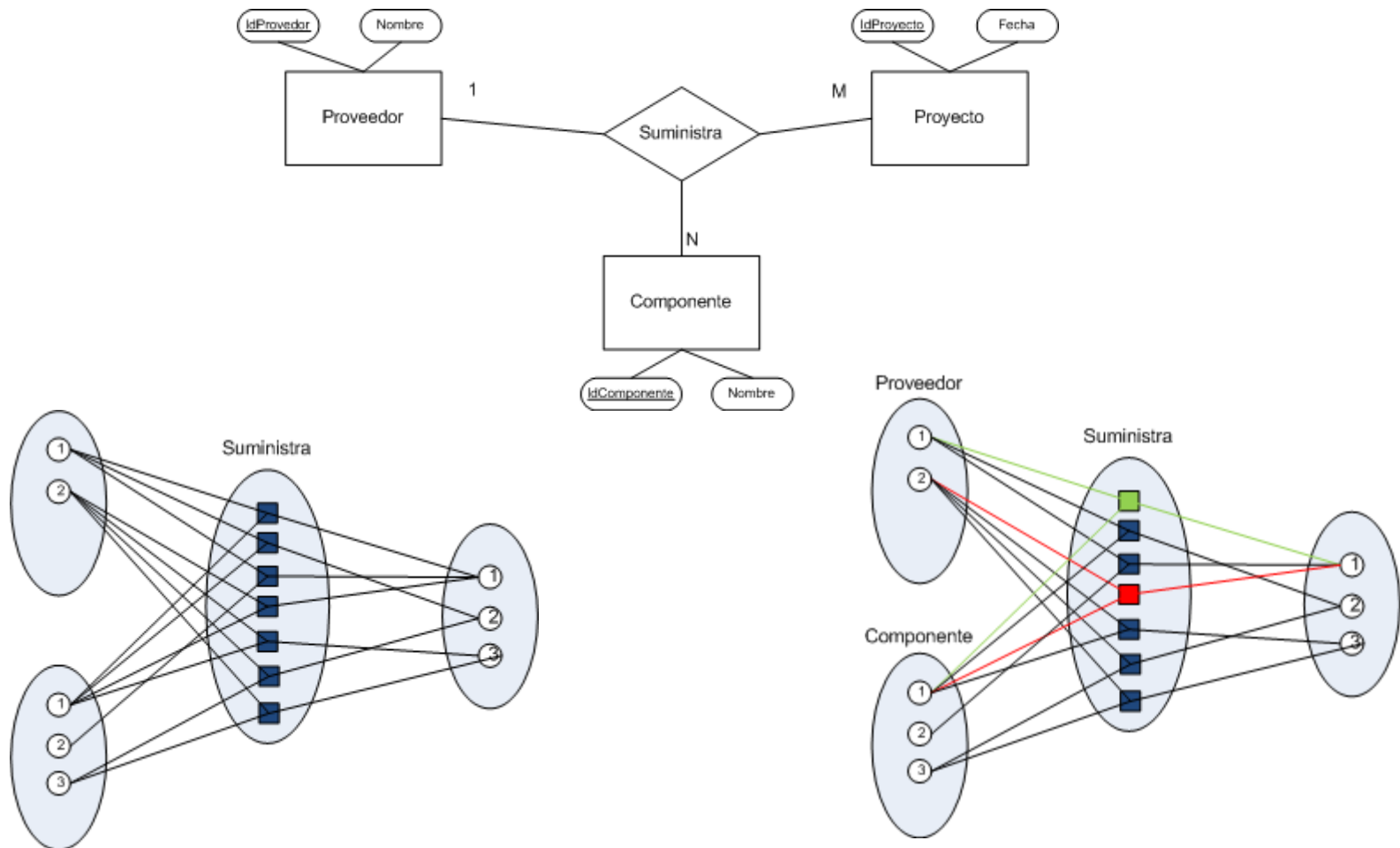


Interrelaciones ternarias

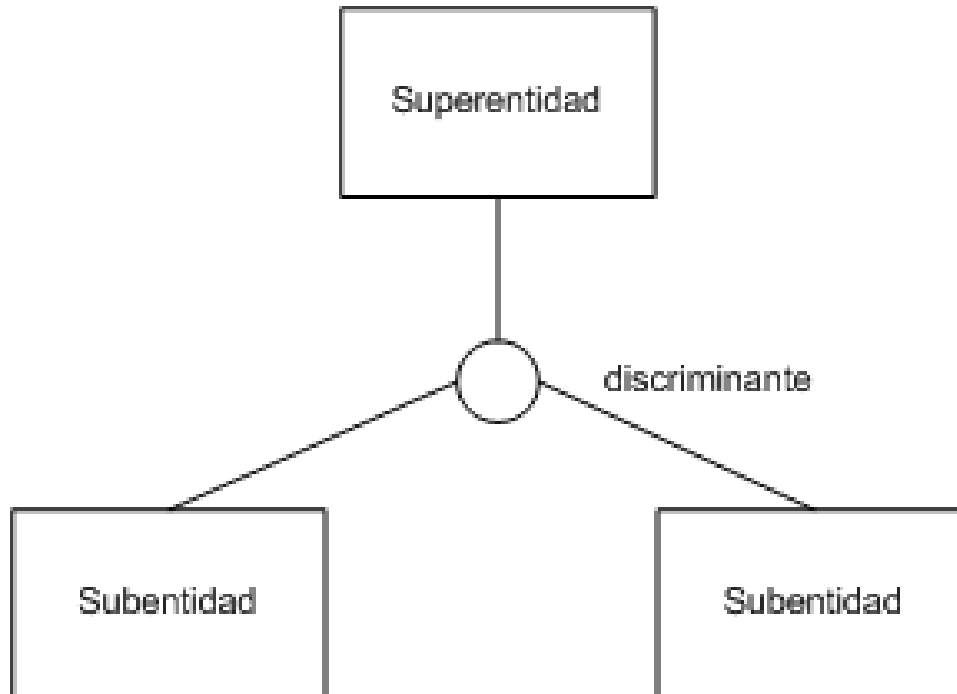
- ▶ Supongamos la interrelación Bebe que relaciona Persona, Bar y Bebida, dicha interrelación tiene un atributo descriptivo *fecha*.
- ▶ ¿Hay alguna combinación de interrelaciones binarias adecuadas para sustituir la ternaria? $\langle P, Ba, Be \rangle$
- ▶ P “gusta” Be, P “visita” Ba, and Ba “sirve” Be no implica que ***P bebe Be en Ba*** ¿Como guardamos la fecha?
- ▶ Una ternaria siempre puede ser reemplazada por un conjunto de binarias. Pero para ello a veces hay que crear otra entidad.



Interrelaciones ternarias

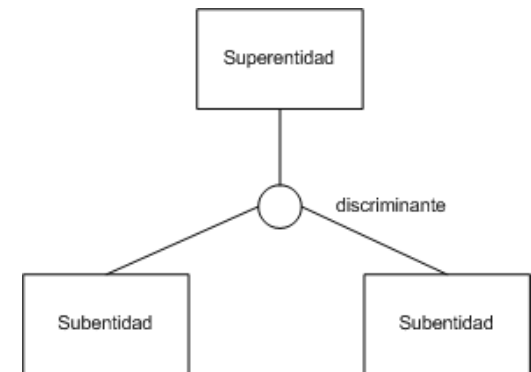
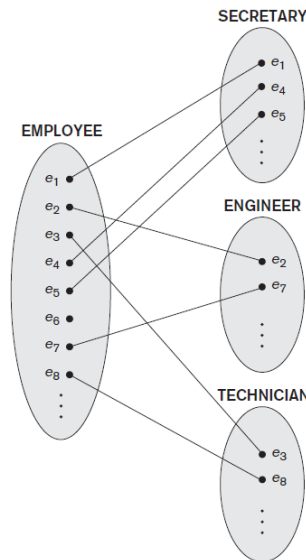


Jerarquías

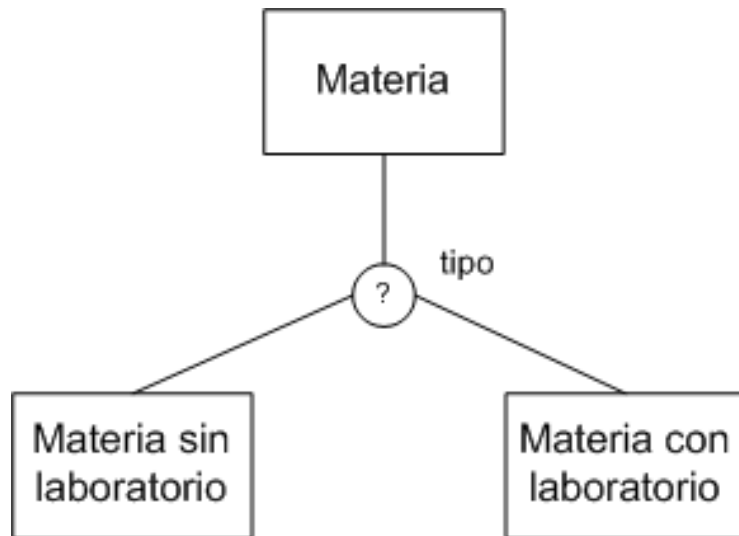


Jerarquías

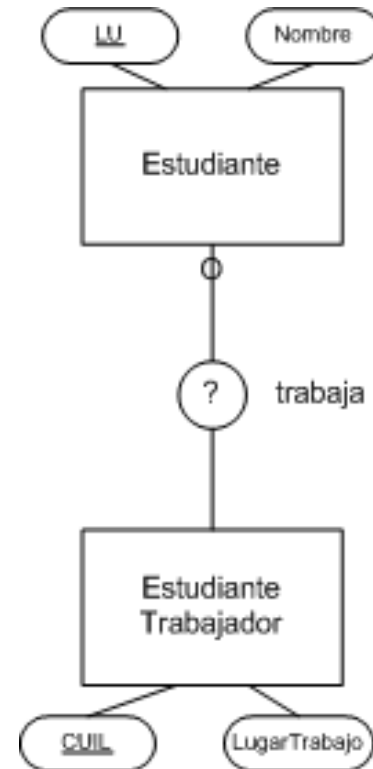
- ▶ Entidad Padre y una o más entidades hijas.
- ▶ Las entidades hijas o *subentidades* heredan la clave de la *superentidad* o entidad padre.
- ▶ Relación es_un
- ▶ Cobertura
 - ▶ Total
 - ▶ Parcial
- ▶ Solapamiento
 - ▶ Disjuntas
 - ▶ Con solapamiento



Jerarquías-Coberturas

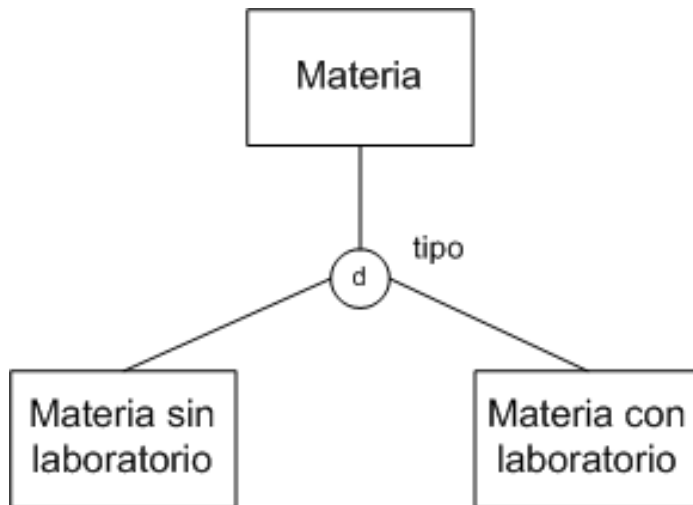


TOTAL

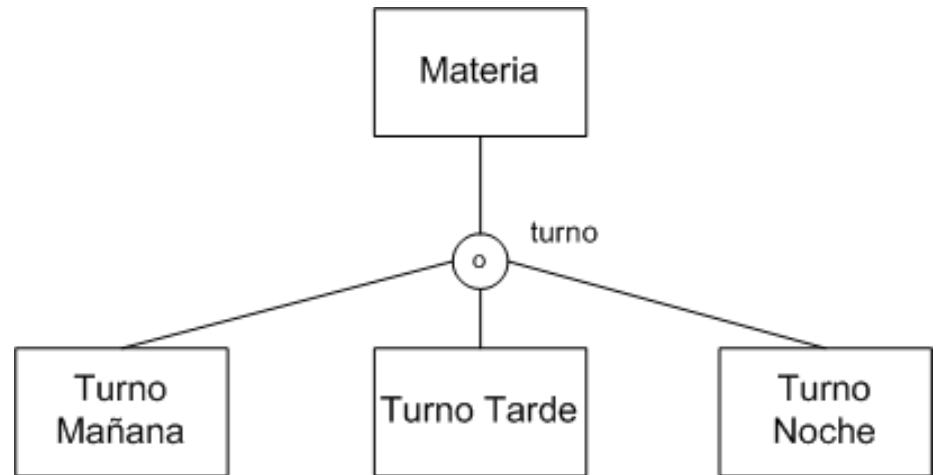


PARCIAL

Jerarquías -Solapamiento



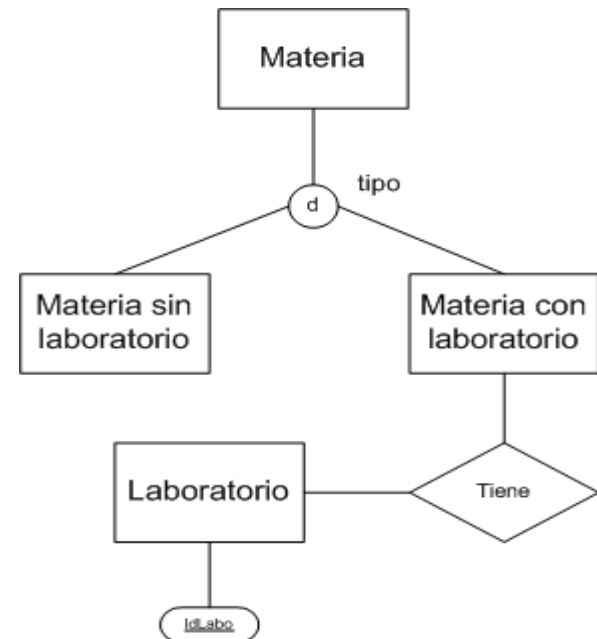
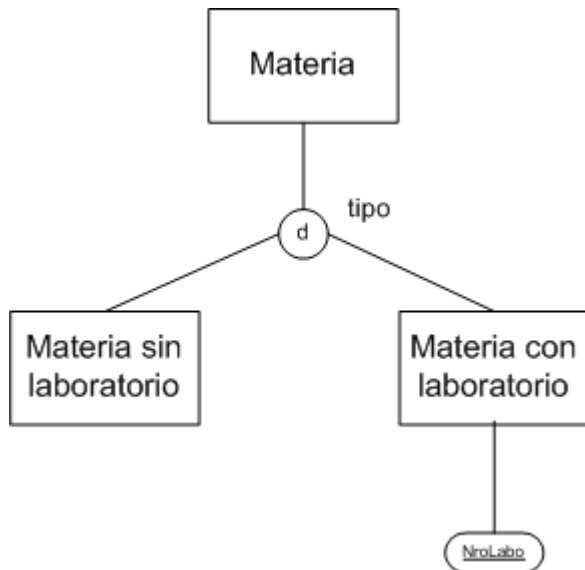
Disjunta



Con solapamiento

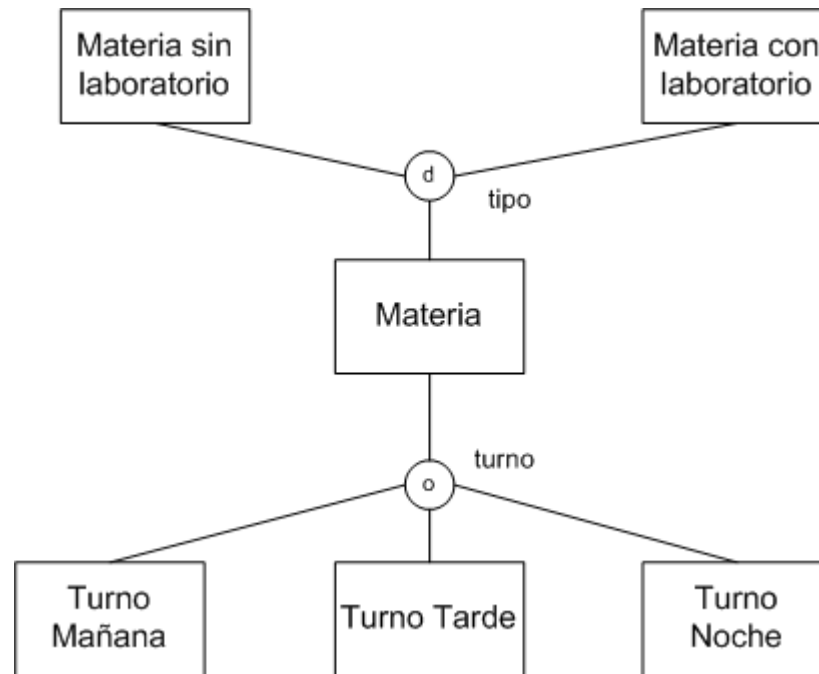
Jerarquías

- La definición de una jerarquía puede estar guiada por la *existencia de atributos distintos* y también por la *existencia de interrelaciones diferentes*



Jerarquías

- Cada jerarquía tiene una semántica. Es posible que una misma entidad tenga diferentes especializaciones con distinta semántica.



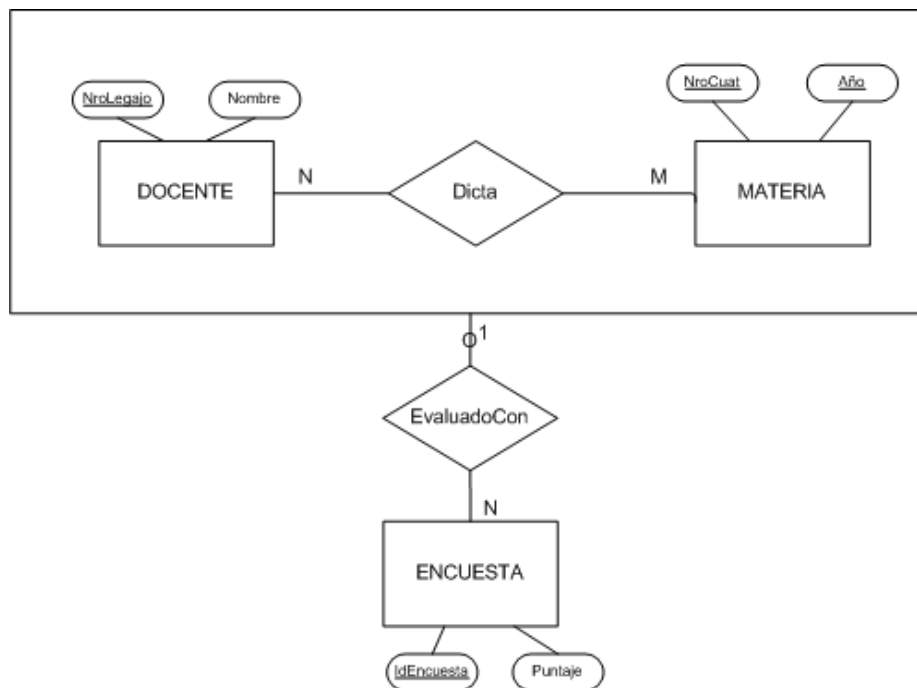
Agregación

- ▶ Se registran datos de los docentes y de las materias dictadas. Una materia puede ser dictada por muchos docentes y un docente puede dictar muchas materias. Además, en algunos cursos (no en todos) se realizan encuestas de evaluación docente. Cada encuesta corresponde a un docente que dicta una materia, y cada docente en una materia puede tener más de una encuesta (o ninguna).
- ▶ ¿Por qué no una ternaria?



Agregación

- La agregación es una abstracción en la cual una interrelación (junto con sus entidades vinculadas) es tratada como una entidad de alto nivel y puede participar de interrelaciones



Diseño de una Base de Datos

► Análisis de Requerimientos

- El objetivo es entender qué es lo que espera el usuario de la BD (qué datos se van a almacenar, identificar operaciones más frecuentes, etc.)

► Diseño Conceptual

- La información recabada se usa para armar un modelo de alto nivel de lo esperado
- Se realiza típicamente usando un MER

► Diseño Lógico

- El modelo de alto nivel se lleva a otro modelo más cercano a la implementación
- Si el motor de BD a utilizar implementa el **modelo relacional**, se utilizará ese modelo

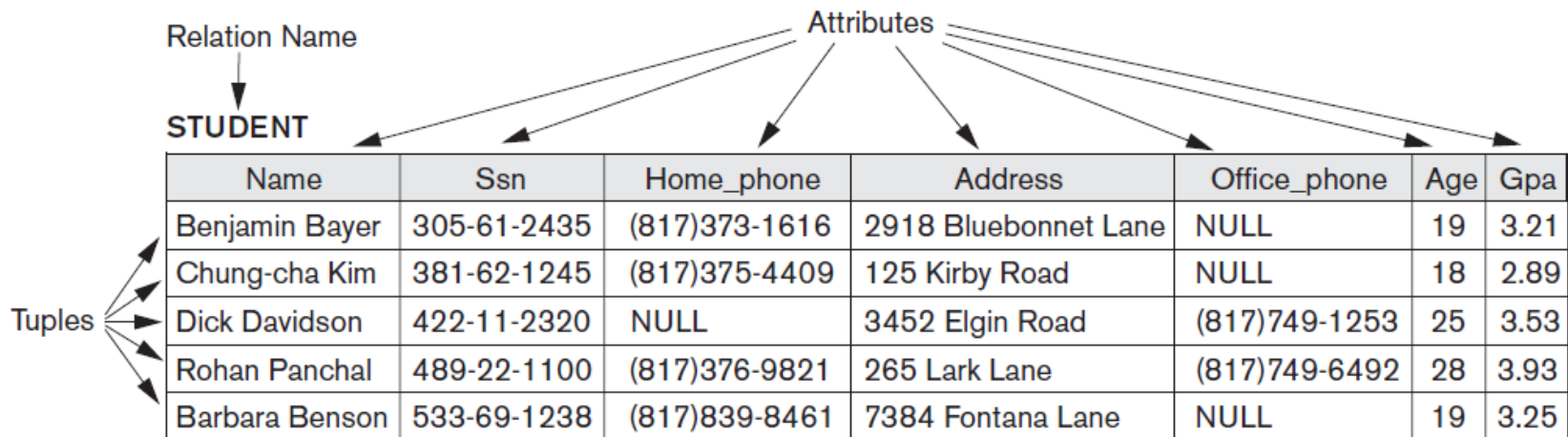
► Refinamiento

- Puede involucrar la normalización o desnormalización de tablas

Modelo Relacional - Definiciones

- ▶ ‘A relational model of data for large shared data banks’ (Codd, 1970).
- ▶ Base de Datos como conjunto de relaciones.
- ▶ Intuitivamente, una relación puede pensarse como una **tabla**, con **filas** y **columnas**
 - ▶ Cada **tabla** es una relación y tiene su nombre
 - ▶ Cada columna de la tabla representa un **atributo**, asociado a un conjunto de valores posibles que puede tomar. A este conjunto es a lo que llamamos **dominio** del atributo
 - ▶ Cada fila, a la que denominaremos **tupla** está formada por un conjunto de valores de datos relacionados

Modelo Relacional



Modelo Relacional - Definiciones

- ▶ Un **dominio D** es un conjunto de valores atómicos. Por lo que respecta al modelo relacional, atómico significa indivisible.
- ▶ Una **relación** se compone de un **esquema** (o intensión de la relación) y una **extensión**,
- ▶ El esquema de la relación consiste en un nombre de relación R y un conjunto de atributos $\{A_1, A_2, \dots, A_n\}$. La **aridad** de una relación es la cantidad de atributos que tiene.
- ▶ Un atributo A_i es el nombre del papel que ejerce algún dominio D en un esquema de relación. D es el dominio de A_i y se denota como **dom(A_i)**.

Modelo Relacional - Definiciones

- ▶ La extensión de la relación de esquema $R(A_1, A_2, \dots, A_n)$ denotada como $r(R)$ es un **conjunto de tuplas** t_i ($i = 1, 2, \dots, m$), donde cada tupla t_i es, a su vez un conjunto de pares $t_i = \{ \langle A_1:v_{i1} \rangle, \langle A_2:v_{i2} \rangle \dots \langle A_n:v_{in} \rangle \}$ y, para cada par $\langle A_j:v_{ij} \rangle$, se cumple que v_{ij} es un valor de dominio (A_j), o bien un valor especial que denominaremos nulo.
- ▶ En esta definición de tupla no importaría el orden. Podríamos definir también si tomamos el orden de los atributos como relevante a una tupla como $t = \langle v_1, v_2, \dots, v_n \rangle$, donde cada v_i representa un valor del atributo A_i
- ▶ Notar entonces que extensión de una relación es un subconjunto del producto cartesiano de una lista de dominios.
 - ▶ $r(R) \subseteq (dom(A_1) \times dom(A_2) \times \dots \times dom(A_n))$

Relation Name		Attributes						
STUDENT								
		Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Tuples		Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
		Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
		Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
		Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
		Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

Modelo Relacional - Claves

- ▶ En este modelo, aparece un concepto importante
 - ▶ Clave: conjunto minimal de atributos que definen unívocamente a las tuplas
 - ▶ Sea K una clave, E una relación y e_i, e_j tuplas
 - ▶ $\forall e_i, e_j \in E: e_i.K = e_j.K \rightarrow e_i = e_j$
- ▶ Las relaciones pueden tener varias claves
 - ▶ A las claves de la relación se las denomina *Claves Candidatas* (*CandidateKeys, CK*)
 - ▶ Una de ellas será elegida como *Clave Primaria* (*PrimaryKey, PK*)
- ▶ A su vez, pueden referenciar a claves de otras relaciones
 - ▶ Se conocen como *Claves Externas* (*ForeignKeys, FK*)

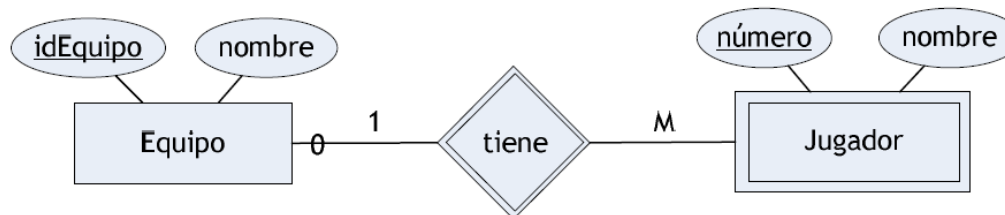
Transformación del MER al MR

► Entidades Fuertes



Estudiante(LU, Nombre, Apellido)

► Entidades Débiles



Equipo(idEquipo, nombre)

Jugador(idEquipo, número, nombre)

Equipo

PK= CK={idEquipo}

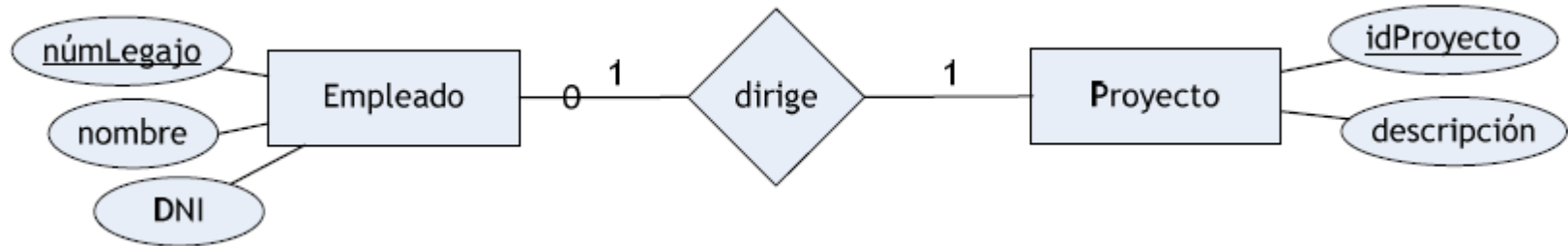
Jugador

PK= CK = {(idEquipo,número)}

FK = {idEquipo}

Transformación del MER al MR

► Interrelaciones 1 : 1



Empleado(númLegajo, nombre, DNI)

PK = {númLegajo}

CK = {númLegajo, DNI}

Proyecto(idProyecto, descripción, númLegajo)

PK = CK = {idProyecto}

FK = {númLegajo}

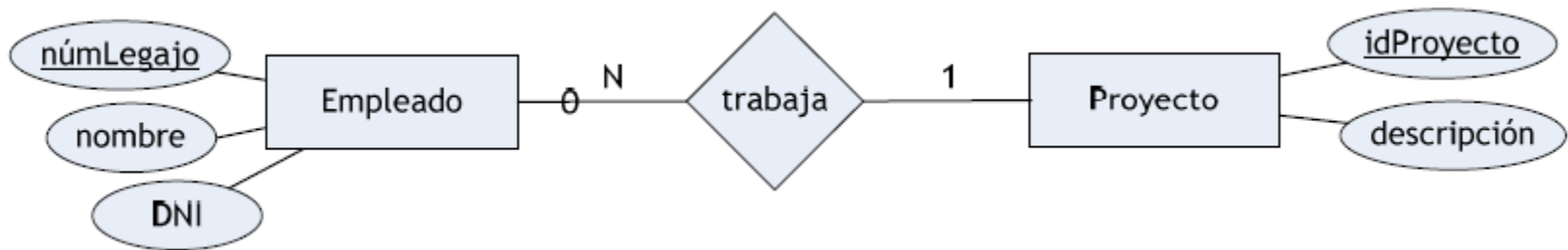
Establecemos las siguientes **restricciones adicionales**:

Empleado.númLegajo puede no estar en Proyecto.númLegajo

Proyecto.númLegajo debe estar en Empleado.númLegajo

Transformación del MER al MR

► Interrelaciones 1:N



Empleado(númLegajo, nombre, DNI, idProyecto)

PK = {númLegajo}

CK = {númLegajo, DNI}

FK = {idProyecto}

Proyecto(idProyecto, descripción)

PK = CK = {idProyecto}

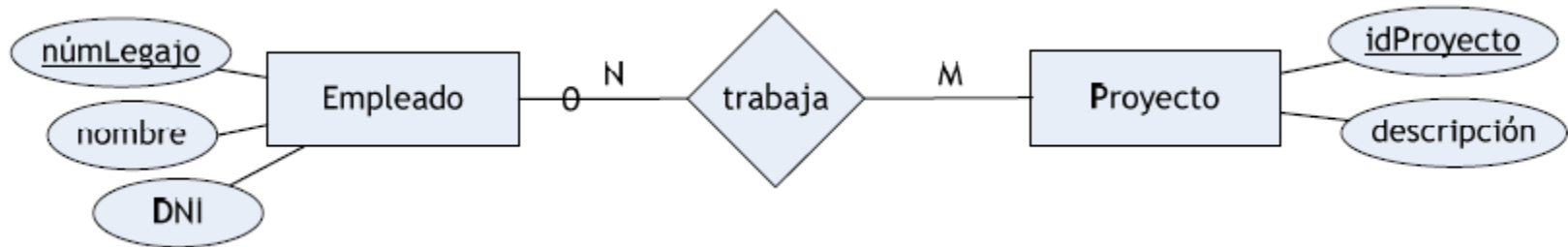
Establecemos las siguientes **restricciones adicionales**:

Empleado.idProyecto es nulo o está en *Proyecto.idProyecto*

Proyecto.idProyecto debe estar en *Empleado.idProyecto*

Transformación del MER al MR

► Interrelaciones N:M



Empleado(númLegajo, nombre, DNI)

PK = {númLegajo}

CK = {númLegajo, DNI}

Proyecto(idProyecto, descripción)

PK = CK = {idProyecto}

Trabaja(númLegajo, idProyecto)

PK = CK = {(númLegajo, idProyecto)}

FK = {númLegajo, idProyecto}

Establecemos las siguientes **restricciones adicionales**:

Empleado.númLegajo puede no estar en *Trabaja.númLegajo*

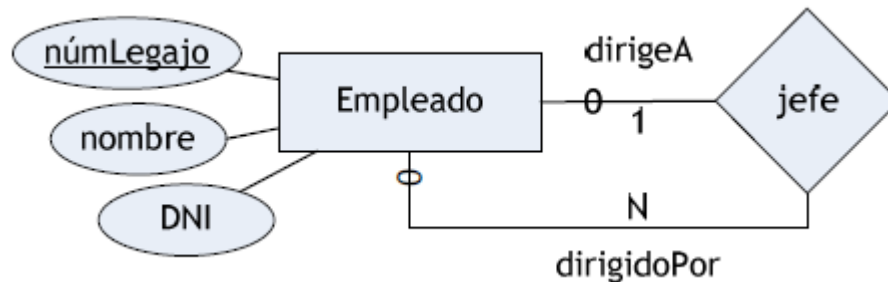
Proyecto.idProyecto debe estar en *Trabaja.idProyecto*

Trabaja.númLegajo debe estar en *Empleado.númLegajo*

Trabaja.idProyecto debe estar en *Proyecto.idProyecto*

Transformación del MER al MR

► Interrelaciones Unarias



Empleado(númeroLegajo, nombre, DNI, númeroLegajoJefe)

PK = {númeroLegajo}

CK = {númeroLegajo, DNI}

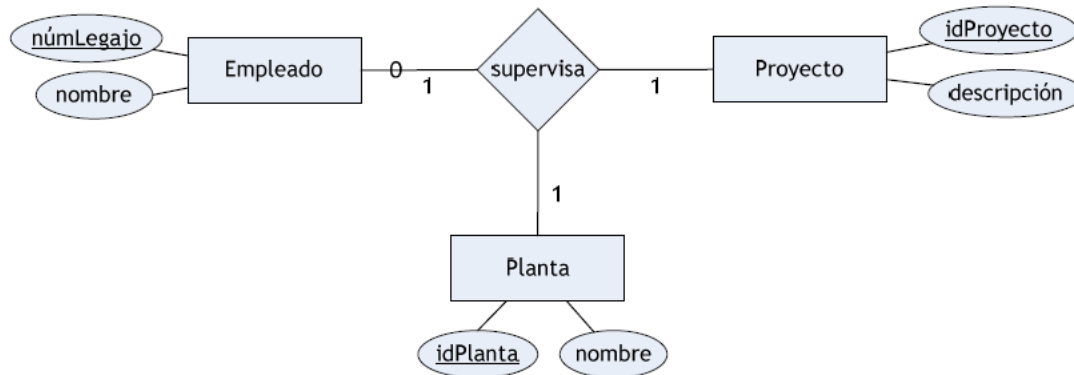
FK = {númeroLegajoJefe}

Empleado.númeroLegajoJefe puede ser nulo o debe estar en Empleado.númeroLegajo

Empleado.númeroLegajo puede no estar en Empleado.númeroLegajoJefe

Transformación del MER al MR

► Ternarias 1:1:1



Empleado(númLegajo, nombre)

PK = CK = {númLegajo}

Proyecto(idProyecto, descripción)

PK = CK = {idProyecto}

Planta(idPlanta, nombre)

PK = CK = {idPlanta}

Supervisa(númLegajo, idProyecto, idPlanta)

PK = {(númLegajo, idProyecto)}

CK = {(númLegajo, idProyecto), (númLegajo, idPlanta), (idProyecto, idPlanta)}

FK = {númLegajo, idProyecto, idPlanta}

Supervisa.númLegajo debe estar en Empleado.númLegajo

Supervisa.idProyecto debe estar en Proyecto.idProyecto

Supervisa.idPlanta debe estar en Planta.idPlanta

Empleado.númLegajo puede no estar en Supervisa.númLegajo

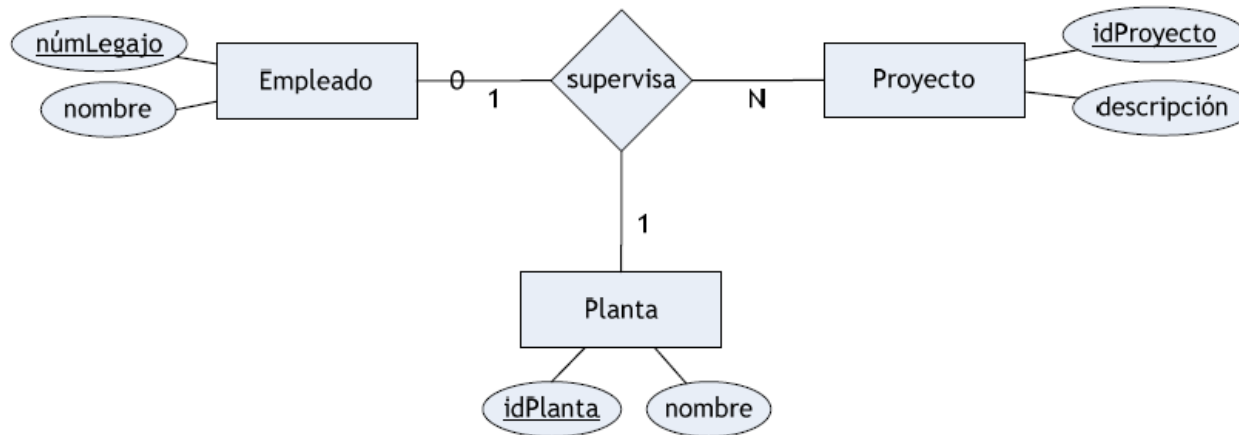
Proyecto.idProyecto debe estar en Supervisa.idProyecto

Planta.idPlanta debe estar en Supervisa.idPlanta

Transformación del MER al MR

► Ternarias 1:1:N

- Cambia el problema y ahora un empleado en una planta puede supervisar varios proyectos



Empleado(númLegajo, nombre)

PK = CK = {númLegajo}

Proyecto(idProyecto, descripción)

PK = CK = {idProyecto}

Planta(idPlanta, nombre)

PK = CK = {idPlanta}

Supervisa(númLegajo, idProyecto, idPlanta)

PK = {(númLegajo, idProyecto)}

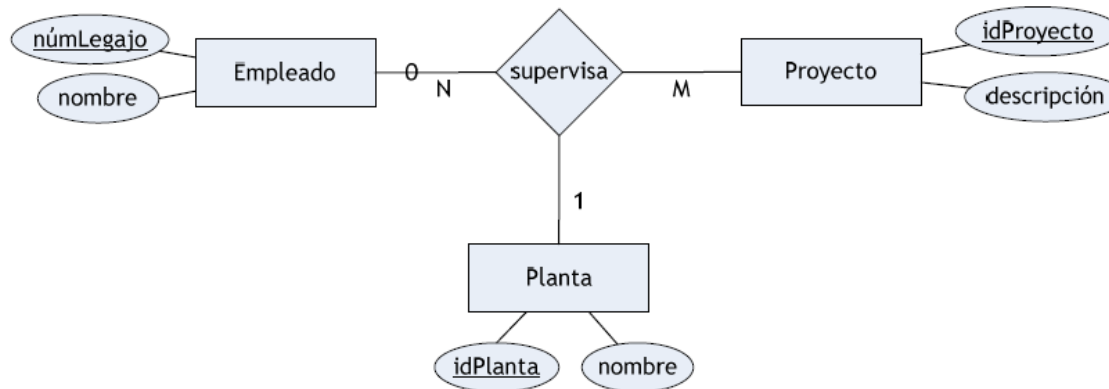
CK = {(númLegajo, idProyecto), (idProyecto, idPlanta)}

FK = {númLegajo, idProyecto, idPlanta}

Transformación del MER al MR

► Ternarias 1:N:M

- Cambia el problema y ahora un proyecto en una planta puede ser supervisado por varios empleados



Empleado(númLegajo, nombre)

PK = CK = {númLegajo}

Proyecto(idProyecto, descripción)

PK = CK = {idProyecto}

Planta(idPlanta, nombre)

PK = CK = {idPlanta}

Supervisa(númLegajo, idProyecto, idPlanta)

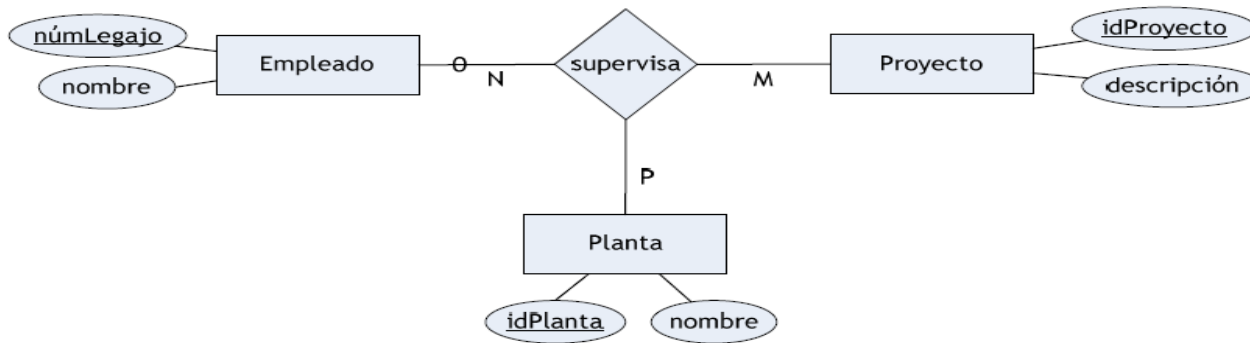
PK = CK = {(númLegajo, idProyecto)}

FK = {númLegajo, idProyecto, idPlanta}

Transformación del MER al MR

► Ternarias N:M:P

- Cambia el problema y ahora un empleado puede supervisar un proyecto en más de una planta.



Empleado(númLegajo, nombre)

PK = CK = {númLegajo}

Proyecto(idProyecto, descripción)

PK = CK = {idProyecto}

Planta(idPlanta, nombre)

PK = CK = {idPlanta}

Supervisa(númLegajo, idProyecto, idPlanta)

PK = CK = {(númLegajo, idProyecto, idPlanta)}

FK = {númLegajo, idProyecto, idPlanta}

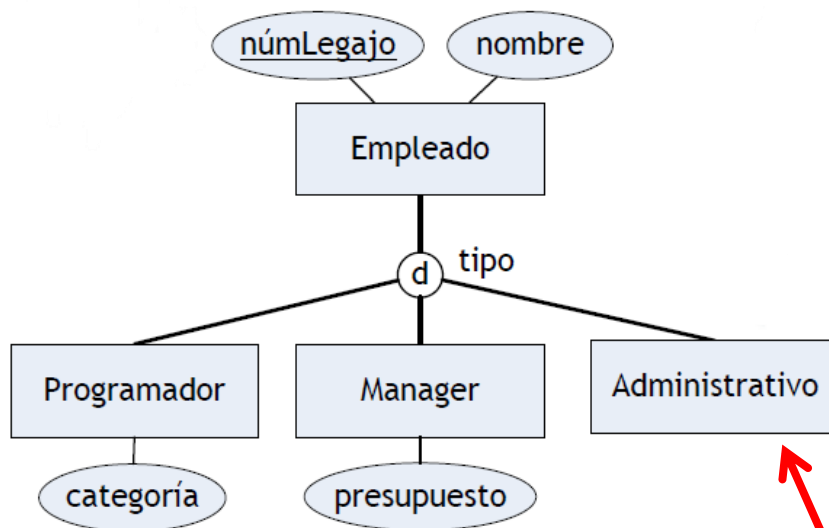
Transformación del MER al MR

▶ RESUMEN TERNARIAS

- ▶ Siempre se genera un esquema aparte para la interrelación.
- ▶ La clave del esquema dependerá de la cardinalidad.

Transformación del MER al MR

► Jerarquías- Disjunta



Se le agrega el discriminante como atributo, es el que permite particionar el conjunto de empleados

Empleado(númLegajo, nombre, tipo)
PK = CK = {númLegajo}

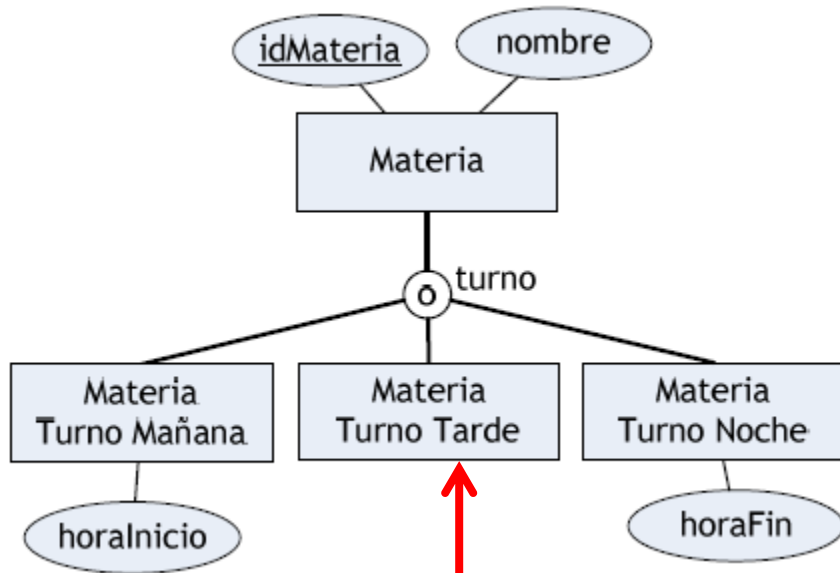
Programador(númLegajo, categoría)
PK = CK = FK = {númLegajo}

Manager(númLegajo, presupuesto)
PK = CK = FK = {númLegajo}

No tiene atributos ni relaciones no es necesario generar un esquema

Transformación del MER al MR

► Jerarquías con solapamiento



No se agrega el discriminante

Materia(idMateria, nombre)
PK = CK = {idMateria}

MateriaTurnoMañana(idMateria, horaInicio)
PK = CK = FK = {idMateria}

MateriaTurnoTarde(idMateria)
PK = CK = FK = {idMateria}

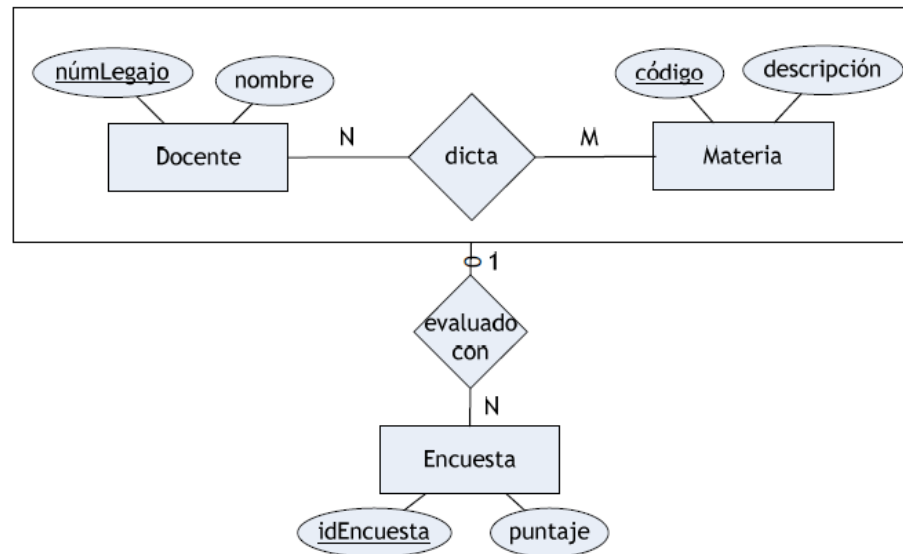
MateriaTurnoNoche(idMateria, horaFin)
PK = CK = FK = {idMateria}

**En este caso si es necesario
generar un esquema**

Transformación del MER al MR

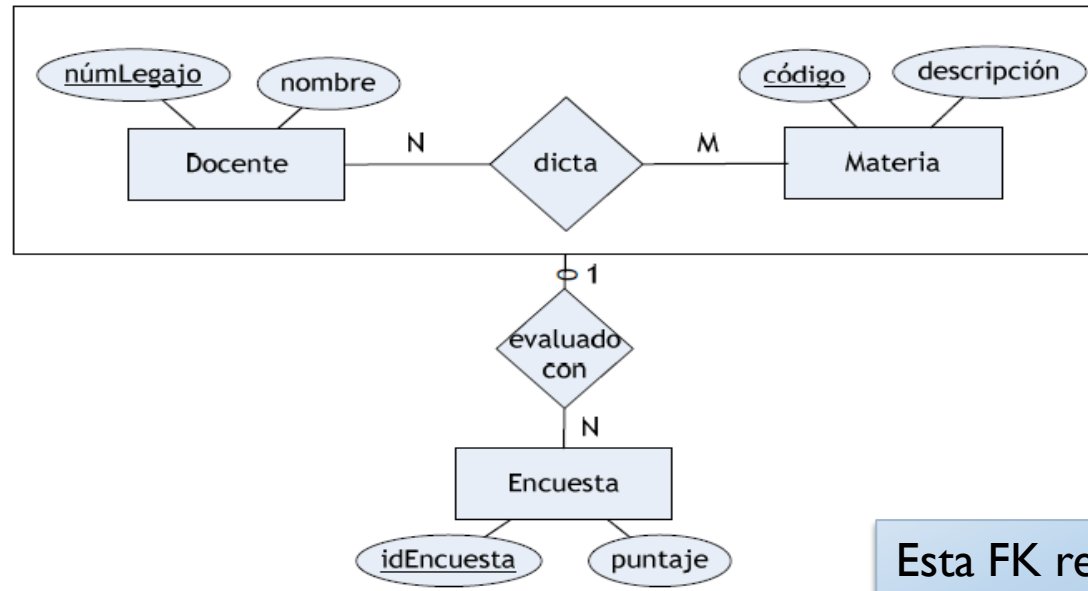
► Agregación

- En la notación de la materia sólo permitimos agregación en interrelaciones N:M.
- Las agregaciones se transforman considerando a la agregación como si fuera una entidad.



Transformación del MER al MR

► Agregación



Esta FK referencia a Dicta

Docente(númLegajo, nombre)
PK = CK = {númLegajo}

Materia(código, descripción)
PK = CK = {código}

Dicta(númLegajo, código)
PK = CK = {(númLegajo, código)}
FK = {númLegajo, código}

Encuesta(idEncuesta, puntaje, númLegajo, código)
PK = CK = {idEncuesta}
FK = {(númLegajo, código)}



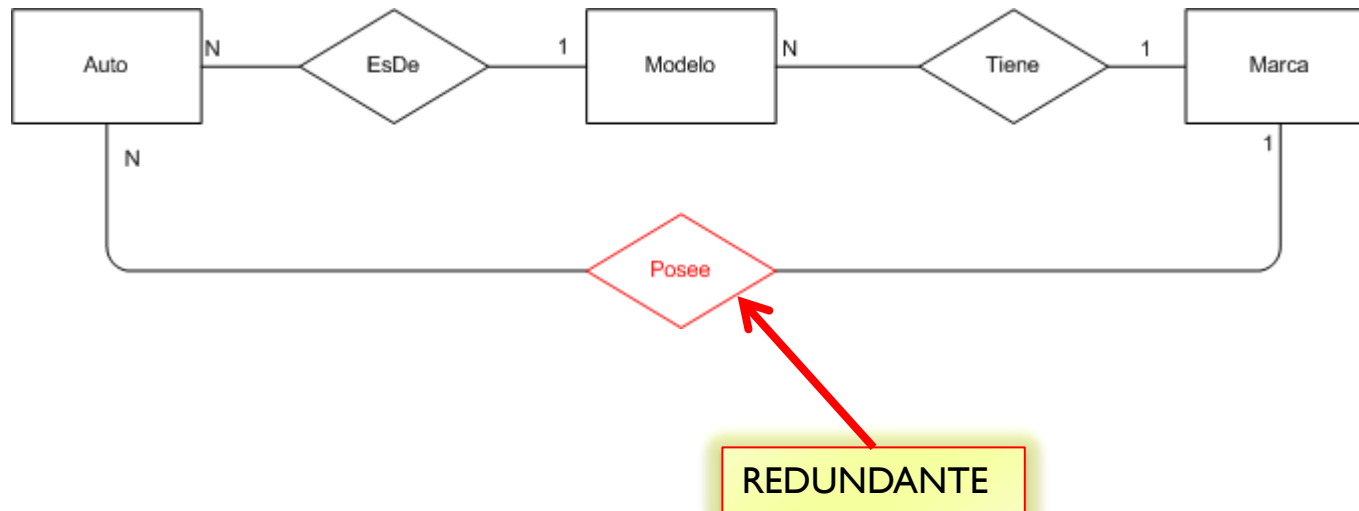


Consideraciones de Diseño



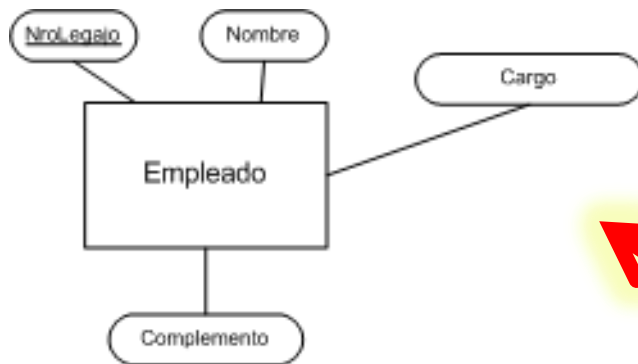
Consideraciones de Diseño

- ▶ El modelo debe respetar fielmente los requerimientos de la situación real que estamos modelando
- ▶ Evitar Redundancia



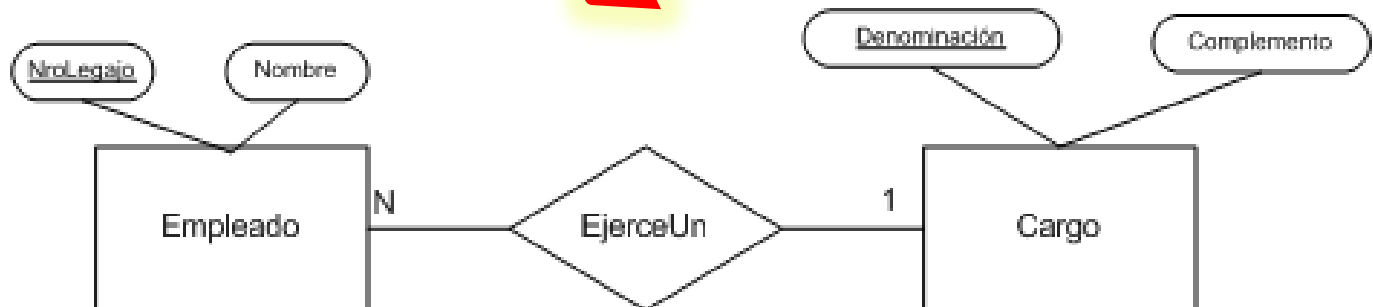
Consideraciones de Diseño

► ¿Entidad o Atributo?



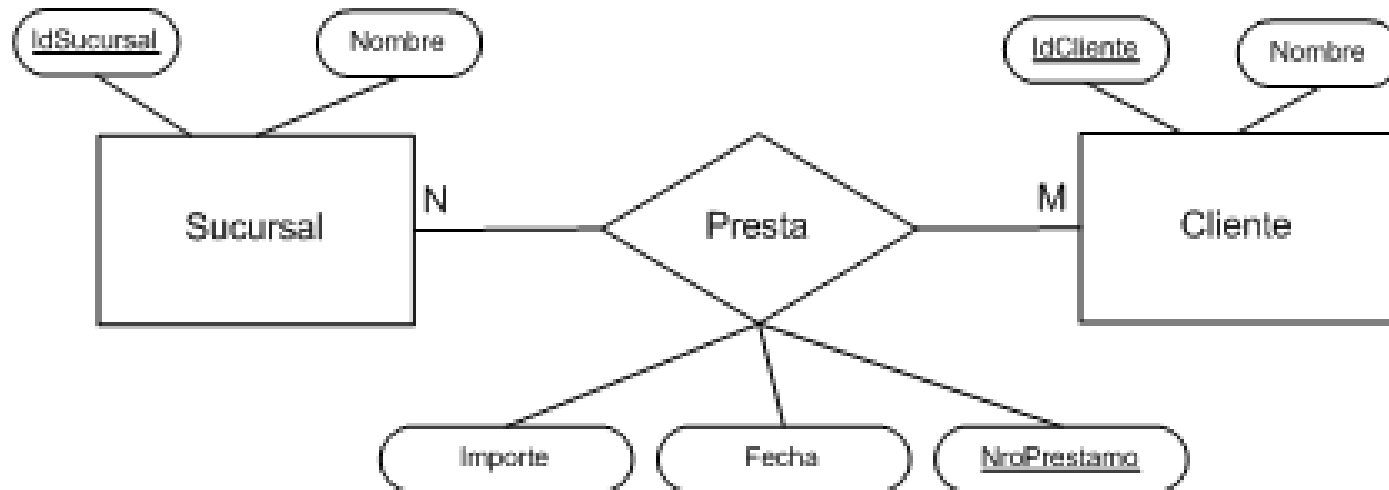
¿Más de un atributo?

¿Posibilidad de otras relaciones?



Consideraciones de Diseño

► ¿Entidad o Interrelación?

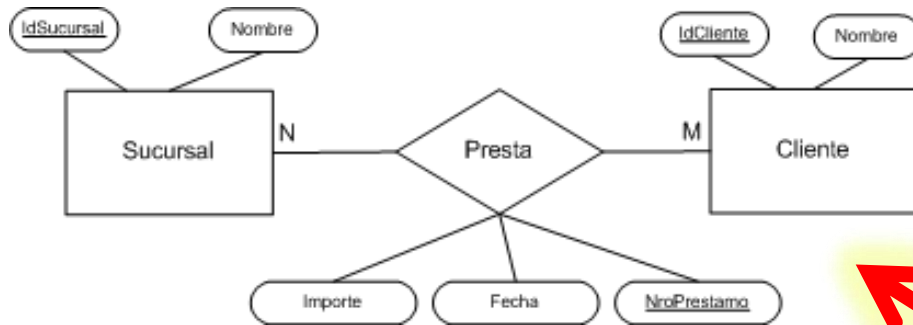


¿Cómo ubicamos los pagos?

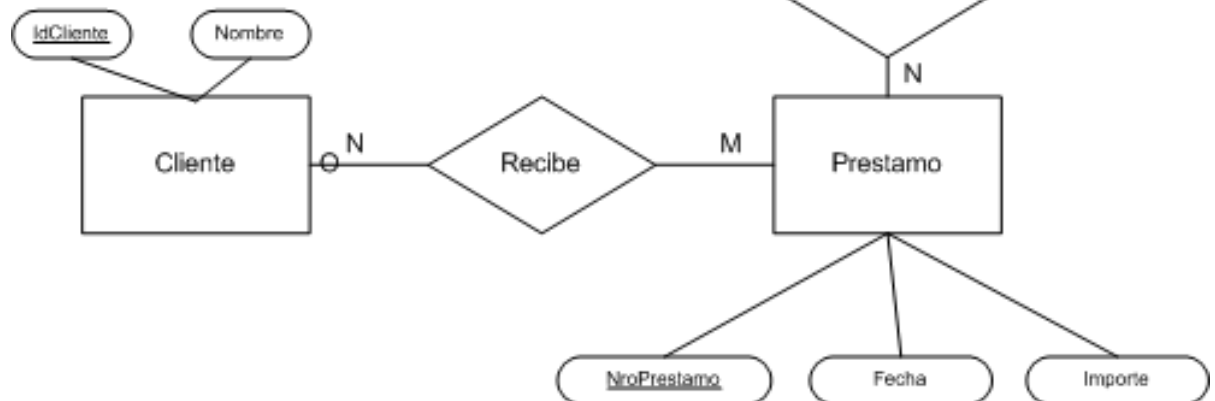
¿Qué pasa si varios clientes comparten un préstamo?

Consideraciones de Diseño

► ¿Entidad o Interrelación?

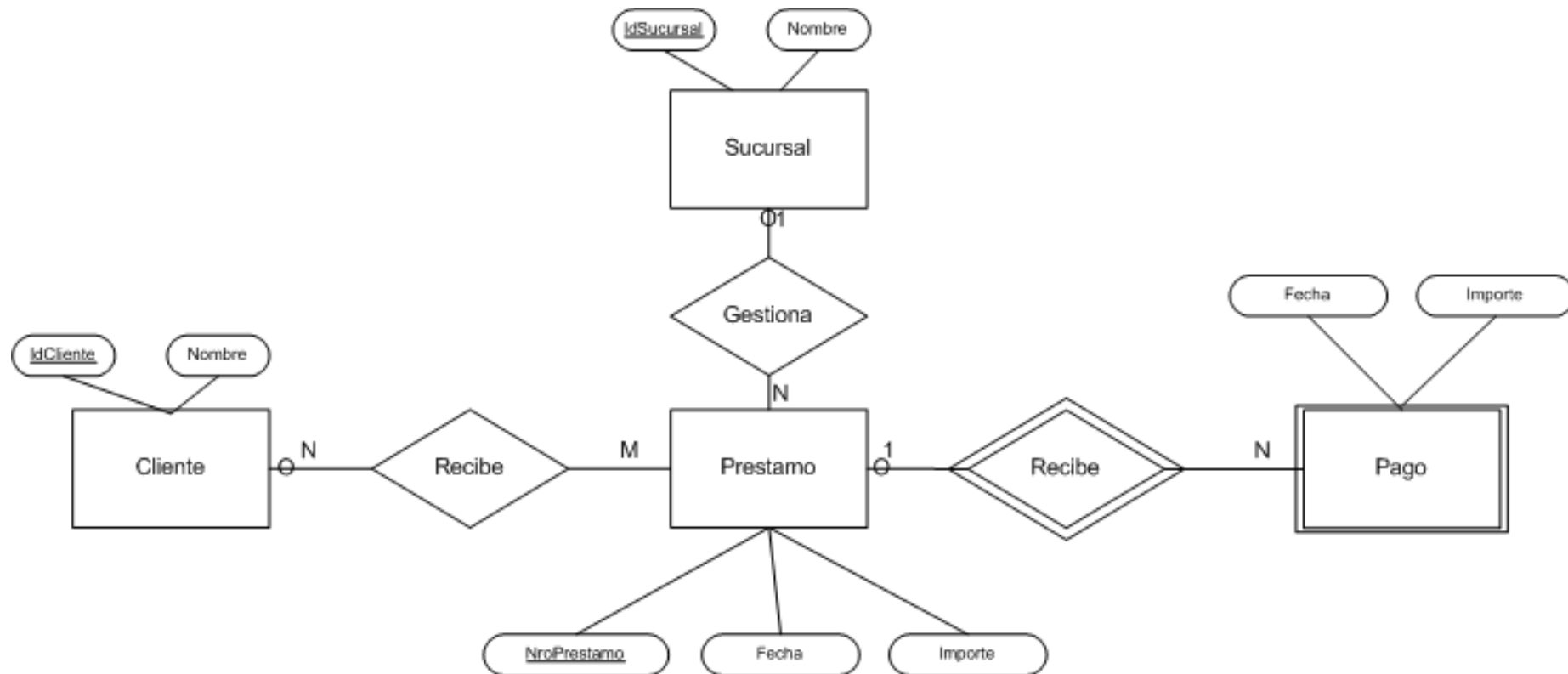


¿Qué pasa si varios clientes comparten un préstamo?



¿Cómo ubicamos los pagos?

Consideraciones de Diseño



¿Cómo ubicamos los pagos?

Consideraciones de Diseño

- ▶ Sustantivos en singular para las entidades
- ▶ Por lo general, usar verbos para las relaciones. Se leen de
 - ▶ Izquierda a derecha
 - ▶ Arriba hacia abajo
- ▶ Elegir buenas Claves
 - ▶ Para elegir una buena clave, debe seleccionarse un atributo que no vaya a cambiar con el tiempo.
 - ▶ Cuando la clave es muy compleja o no existe un atributo que surja del problema real, se recomienda generar una clave artificial (por ejemplo, idEmpleado)

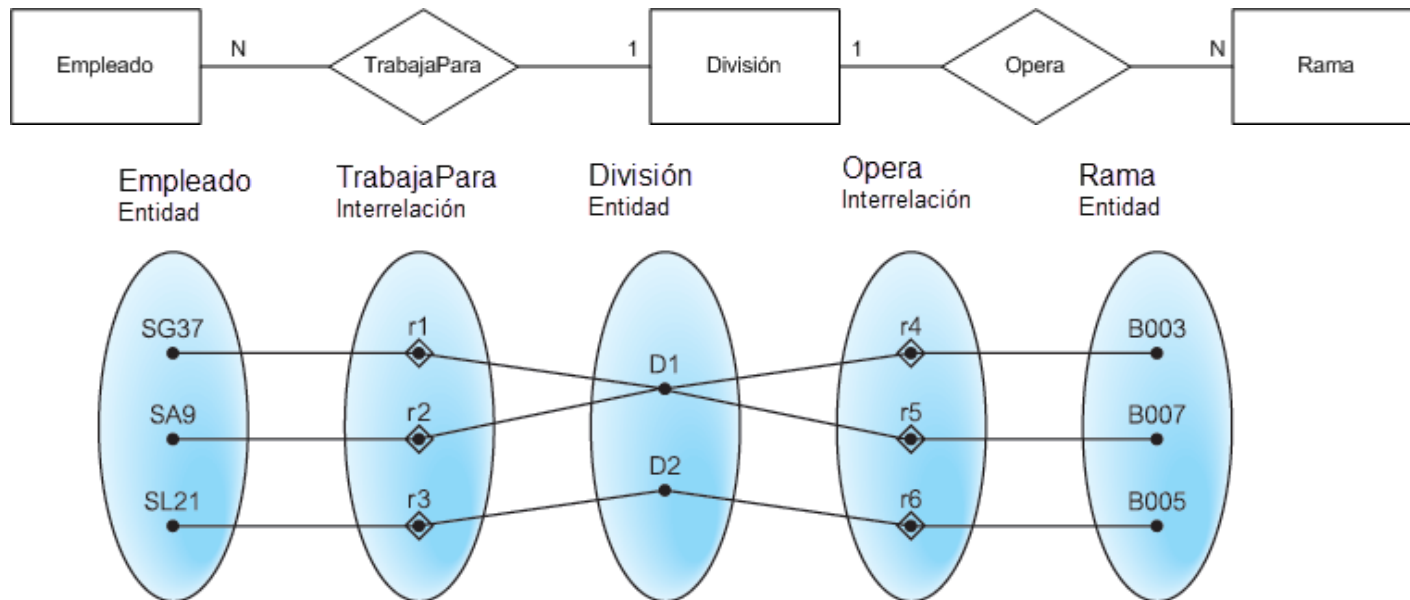
Trampas de Conexión

- ▶ Ocurren debido a una mala interpretación de las interrelaciones.
- ▶ Para identificar las trampas de conexión, debemos asegurarnos de que el **significado** de un tipo de interrelación está completamente **entendido** y claramente **definido**.
- ▶ Si no entendemos las interrelaciones, podríamos crear un modelo que no es una representación adecuada del “mundo real”.
- ▶ ***Fan Traps* o Trampa del Abanico.**
- ▶ ***Chasm Traps* o Trampa del Sumidero.**

Trampas de Conexión

► Trampa del Abanico

El modelo representa una interrelación entre tipos de entidades, pero el camino entre algunas entidades es ambiguo

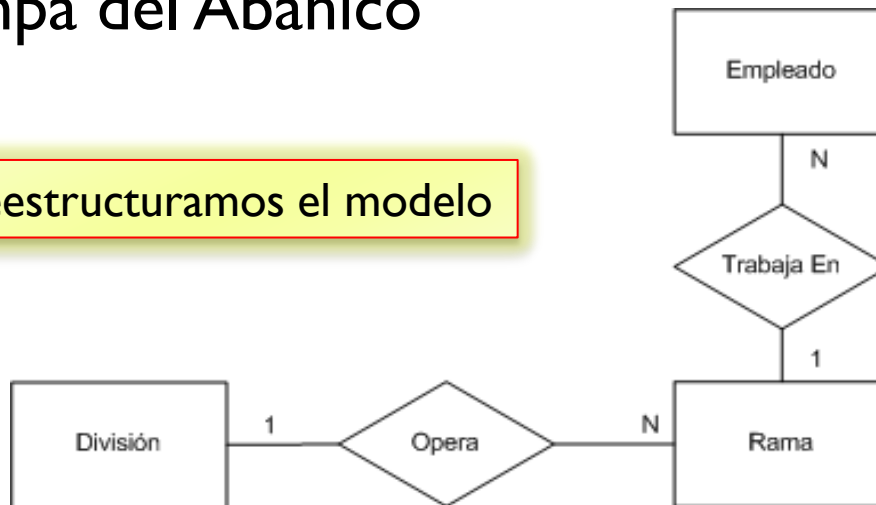


¿Qué empleados trabajan en una rama?

Trampas de Conexión

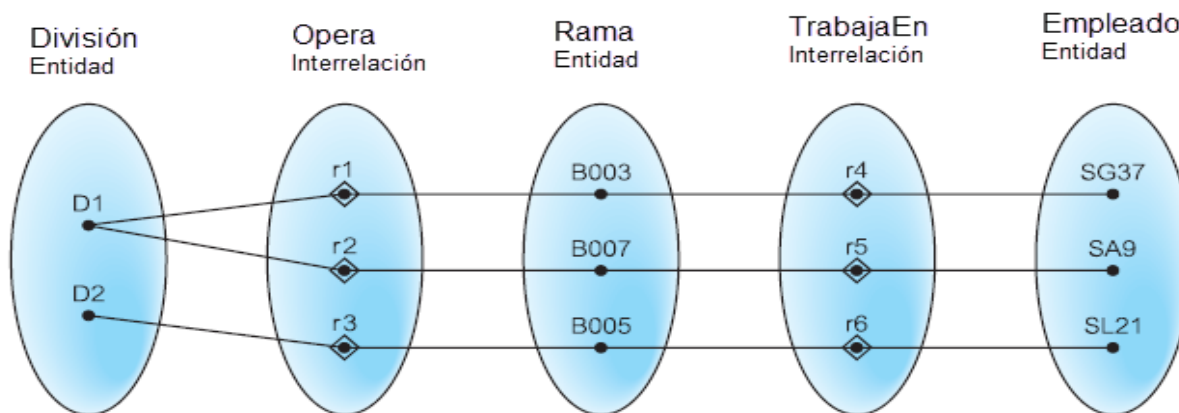
► Trampa del Abanico

Reestructuramos el modelo



¿Qué empleados trabajan en una rama?

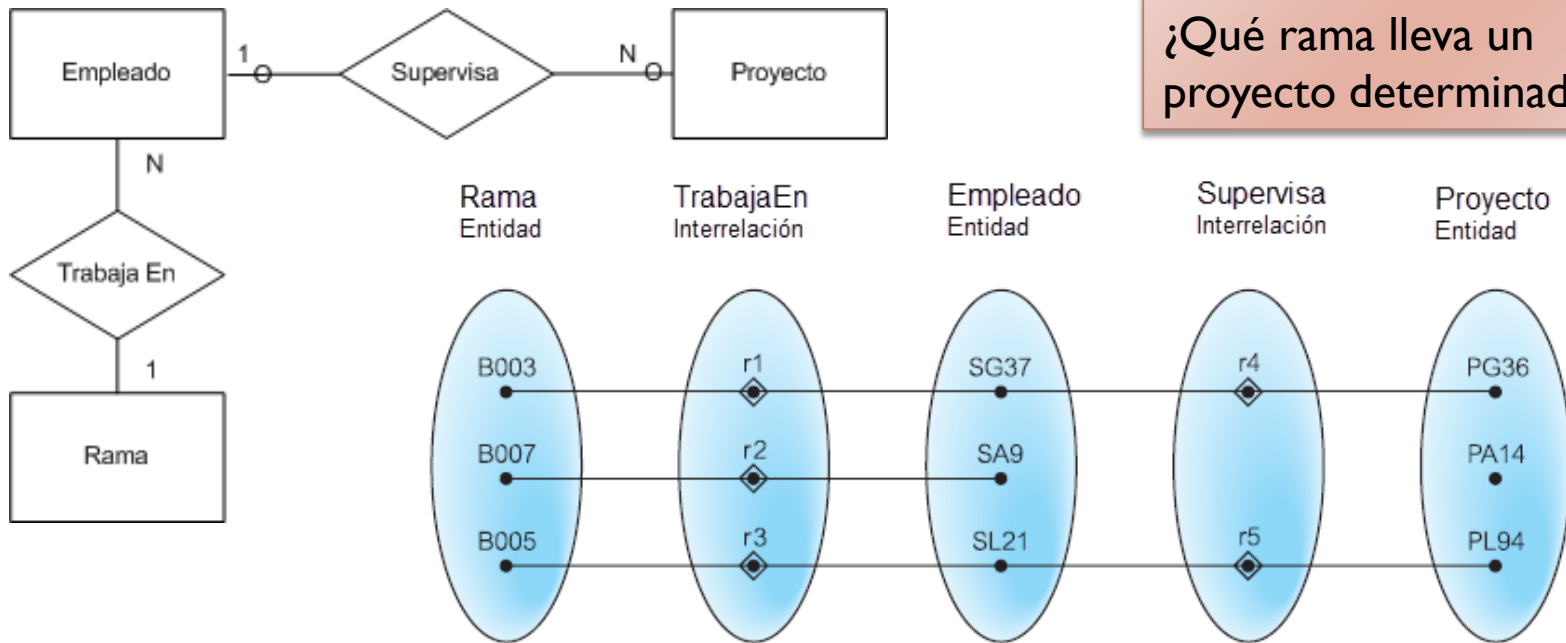
¿Qué empleados trabajan en una división?



Trampas de Conexión

► Trampa del Sumidero

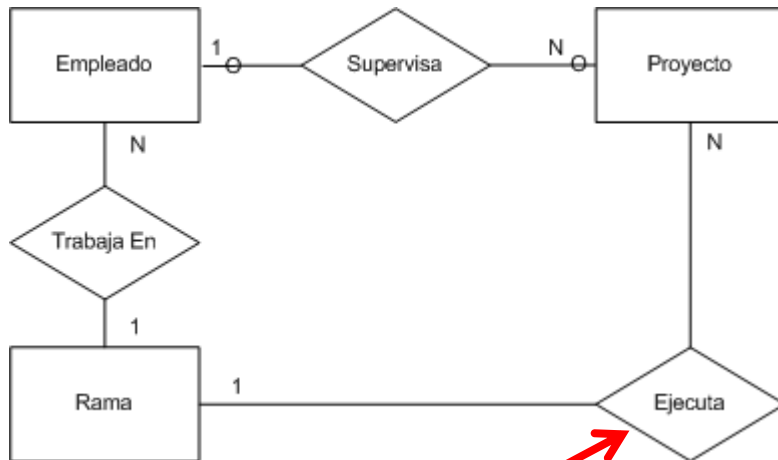
El modelo sugiere la existencia de algún tipo de interrelación entre dos tipos de entidad, pero no existe camino entre algunas entidades.



En una rama trabajan uno o más empleados, los cuales pueden supervisar cero o más proyectos llevados por la rama, que a su vez son supervisados por cero o más empleados.

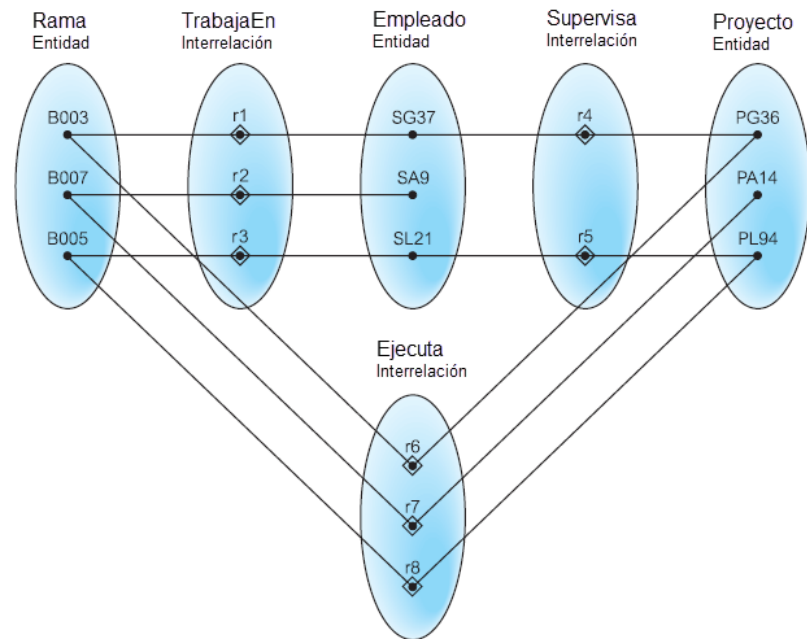
Trampas de Conexión

► Trampa del Sumidero

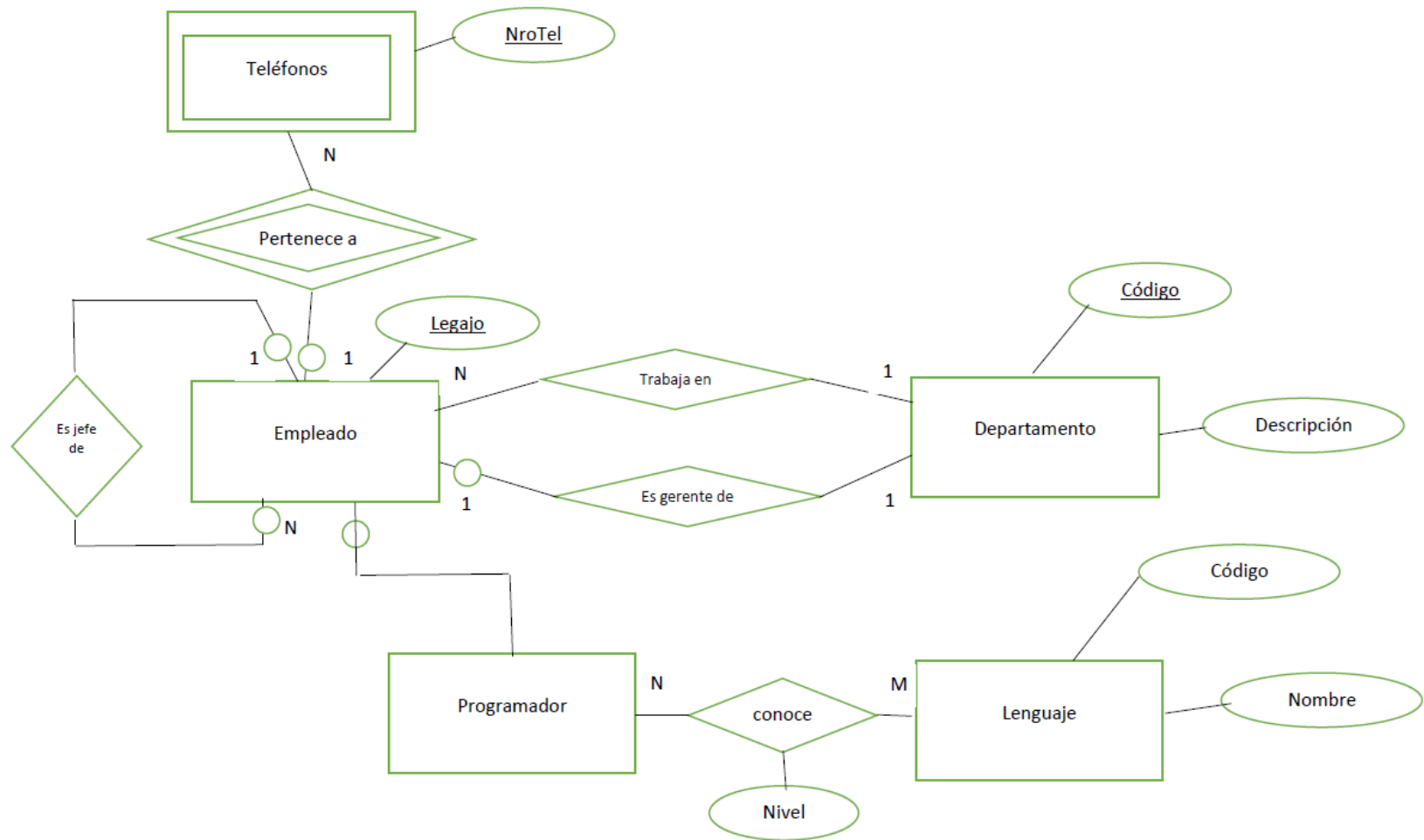


¿Qué rama lleva un proyecto determinado?

Reestructuramos el modelo y Agregamos una Interrelación



Pasar a modelo relacional



Bibliografía

- ▶ Database System Concepts. 4ta Edición Abraham Silberschatz, Henry F. Korth y S. Sudarshan
- ▶ Database Management Systems, Ramakrishnan/Gherke 3rd Ed.
- ▶ A Practical Approach to Design, Implementation, and Management. Thomas Connolly/Carolyn Begg 4ta Edición
- ▶ Apunte de la Materia
- ▶ Fundamentals of Database Systems Elmasri/Navathe 6th Ed., Addison Wesley